Activity: **4.3**
Define Project Requirements

Responsibility: Project Manager/Team

Description: Use the project scope, objectives, and high-level requirements as the basis for defining the project requirements. The questions used to define the project objectives may be helpful in developing the project requirements. The goals for defining project requirements are to identify what functions are to be performed on what data, to produce what results, at what location, and for whom. The requirements must focus on the software products that are needed and the functions that are to be performed. Avoid incorporating design issues and specifications in the requirements.

Requirements should be specified as completely and thoroughly as possible. The requirements must support the system owner's business needs, information resource management long-range plans, and the organizational and Departmental missions. When requirements are being defined, it is not sufficient to state only the requirements for the problems that will be solved; all of the requirements for the project must be captured.

Attributes: Each requirement must be stated as a unique objective with the following attributes. The existence of these attributes must be verified prior to the delivery of the Software Requirements Specification later in the Requirements Definition Stage.

- Necessary - Absolute requirements that are to be verified are indicated by "must" or "shall". Goals or intended functionality are indicated by "will".
- Correct - Each requirement is an accurate description of a feature or process of the software product.
- Unambiguous - The statement of each requirement denotes only one interpretation.
- Complete - Each requirement describes one result that must be achieved by the software product. The requirement should not describe the means of obtaining the result.
- Consistent - Individual requirements are not in conflict with other requirements.

Attributes, continued:

- Verifiable (testable) - Each requirement is stated in concrete terms and measurable quantities. A process should exist to validate that the software product (when developed) will satisfy the set of requirements.
- Modifiable - The structure and style of the requirements are such that any necessary changes to the requirements can be made easily, completely, and consistently.
- Traceable - The origin of each requirement is clear and can be tracked in future development activities and tests.

Identification System:

The creation of a standard identification system for all requirements is required in order to facilitate configuration control, requirements traceability, and testing activities. The identification system must provide a unique designator for each requirement. For example, the identification system can classify the requirements by type (e.g., functional, input, or computer security). Within each type classification, the requirements can be assigned a sequential number. Select an identification system that is appropriate for the scope of the project.

Changes:

As the project evolves, the requirements may change or expand to reflect modifications in the users' business plans, design considerations and constraints, advances in technology, and increased insight into user business processes. A formal change control process must be used to identify, control, track, and report proposed and approved changes. Approved changes in the requirements must be incorporated into the Software Requirements Specification in such a way as to provide an accurate and complete audit trail of the changes. This change control process should be an integral part of the project's Software Configuration Management Plan.

Tasks:

The following tasks are involved in developing project requirements.

4.3.1 Define Functional Requirements
4.3.2 Define Input and Output Requirements
4.3.3 Define Performance Requirements
4.3.4 Define User Interface Requirements
4.3.5 Define System Interface Requirements
4.3.6 Define Communication Requirements
4.3.7 Define Computer Security and Access Requirements
4.3.8 Define Backup and Recovery Requirements
4.3.9 Define Data Requirements
4.3.10 Define Implementation Requirements

Task: **4.3.1**
Define Functional Requirements

Description: Functional requirements define what the software product must do to support the system owner's business functions and objectives. The functional requirements should answer the following questions.

- How are inputs transformed into outputs?
- Who initiates and receives specific information?
- What information must be available for each function to be performed?

Identify requirements for all functions whether they are to be automated or manual. Describe the automated and manual inputs, processing, outputs, and conditions for all functions. Include a description of the standard data tables and data or records that will be shared with other applications. Identify the forms, reports, source documents, and inputs/outputs that the software product will process or produce to help define the functional requirements.

A functional model should be developed to depict each process that needs to be included. The goal of the functional model is to represent a complete top-down picture of the software product.

Flow diagrams should be used to provide a hierarchical and sequential view of the system owner's business functions and the flow of information through the processes.

Work Product: Maintain a record of all functional requirements. Save for incorporation into the Software Requirements Specification. Place a copy of the functional requirements in the Project File.

Sample Functional Requirement: *The selection criteria for the extraction of records shall be the occurrence of the letters "EW" in the Budget and Reporting Code field.*

Optional Work Product: Consider developing an optional work product that defines how the final software product will operate to support the system owner organization's business functions and objectives. This user-oriented requirements manual would identify processes in a narrative form from the user's perspective and would include requirements for all functions whether they are to be automated or manual. A functional description can be developed to depict each process that will be provided. The goal is to present a complete top-down picture of the software product.

Optional Work Product, continued: This user-oriented requirements manual can be used as an aid in validating the functional requirements and serves as the basis for the user documentation. If a test group outside the project team is used, the test group can work with the project team to develop the manual.

Review Process: Conduct structured walkthroughs as needed to ensure the necessity, testability, accuracy, and completeness of the functional requirements.

Task: **4.3.2**
Define Input and Output Requirements

Description: Describe all manual and automated input requirements for the software product such as data entry from source documents and data extracts from other applications.

Describe all output requirements for the software product such as printed reports, display screens, and files.

Work Product: Maintain a record of all input and output requirements. Save for incorporation into the Software Requirements Specification. Place a copy of the input and output requirements in the Project File.

Sample Input Requirement: *The application must automatically assign a unique, sequential Employee Number to each employee record that is entered into the data base.*

Sample Output Requirement: *All reports that contain Privacy Act data must include a warning statement in the report header information.*

Review Process: Conduct structured walkthroughs as needed to ensure the necessity, testability, accuracy, and completeness of the input and output requirements.

Task: **4.3.3**
Define Performance Requirements

Description: Performance requirements define how the software product must function (e.g., hours of operation, response times, and throughput under detailed load conditions). The information gathered in defining the project objectives can translate into very specific performance requirements; (e.g., if work performed for an organization is mission essential to the Department, the hours of operation and throughput will be critical to meeting the mission). Also, Government and DOE policy can dictate specific availability and response times.

Work Product: Maintain a record of all performance requirements. Save for incorporation into the Software Requirements Specification. Place a copy of the performance requirements in the Project File.

Sample Performance Requirement: *The application must be available for use from 8:00 a.m. to 5:00 p.m. Monday through Friday.*

Review Process: Conduct structured walkthroughs as needed to ensure the necessity, testability, accuracy, and completeness of the performance requirements.

Task: **4.3.4**
Define User Interface Requirements

Description: The user interface requirements should describe how the user will access and interact with the software product, and how information will flow between the user and the software product.

Interface Issues: The following are some of the issues that should be considered when trying to identify user interface requirements.

- The users' requirements for screen elements, navigation, and help information.
- The standards for the programmatic organization, DOE, Government, and industry that apply to user interfaces.
- The range of skill levels of the users who will access and use the software product.
- The range of work that the users will be performing with the software product.

Define the user interface requirements by identifying and understanding what is most important to the user, not what is most convenient for the project team.

Work with the system owner and users to develop a set of user interface requirements that can be used for all automated products for the system owner's organization. A standard set of user interface requirements will simplify the design and code processes, and ensure that all automated products have a similar look and feel to the users. When other constraints (such as a required interface with another application) do not permit the use of existing user interface standards, an attempt should be made to keep the user interface requirements as close as possible to the existing standard.

Work Product: Maintain a record of all user interface requirements. Save for incorporation into the Software Requirements Specification. Place a copy of the user interface requirements in the Project File.

Sample User Interface Requirement: *All data entry screens must include a unique screen identification number.*

Review Process: Conduct structured walkthroughs as needed to ensure the necessity, testability, accuracy, and completeness of the user interface requirements.

Task: **4.3.5**

Define System Interface Requirements

Description: The hardware and software interface requirements must specify hardware and software interfaces required to support the development, operation, and maintenance of the software product.

The following information should be considered when defining the hardware and software interface requirements.

- System owner's and users' computing environment.
- Existing or planned software that will provide data to or accept data from the software product.
- Other organizations or users having access to the software product.
- Purpose or mission of interfacing software.
- Common users, data elements, reports, and sources for forms/events/outputs.
- Timing considerations that will influence sharing of data, direction of data exchange, and security constraints.
- Development constraints such as the operating system, data base management system, language compiler, tools, utilities, and network protocol drivers.
- Standardized system architecture defined by hardware and software configurations for organizations, programmatic offices, or telecommunications programs.

Work Product: Maintain a record of all system interface requirements. Save for incorporation into the Software Requirements Specification. Place a copy of the system interface requirements in the Project File.

Sample System Interface Requirement: *The application must extract records with the following position status indicators from the HRIS mainframe application: EN, X, D, P, T or NW.*

Review Process: Conduct structured walkthroughs as needed to ensure the necessity, testability, accuracy, and completeness of the system interface requirements.

Task: **4.3.6**
Define Communication Requirements

Description: The communication requirements define connectivity and access requirements within and between user locations and between other groups and applications.

The following factors should be considered when defining communication requirements.

- Communication needs of the user and customer organizations.
- User organization's existing and planned telecommunications environment (e.g., LANs, WANs, and dial-up).
- Projected changes to the current communication architecture, such as the connection of additional local and remote sites.
- Limitations placed on communications by existing hardware and software including:
  - existing user systems
  - existing applications that will interface with the software product
  - existing organizations that will interface with the software product
- Organization, Government, and industry standards that define communication requirements and limitations

Work Product: Maintain a record of all communication requirements. Save for incorporation into the Software Requirements Specification. Place a copy of the communication requirements in the Project File.

Sample Communication Requirement: *The application must execute online in the organization's local area network environment.*

Review Process: Conduct structured walkthroughs as needed to ensure the necessity, testability accuracy, and completeness of the communications requirements.

Task: **4.3.7**
Define Computer Security and Access Requirements

Description: Develop the computer security requirements in conjunction with the system owner's Computer System Security Officer (CSSO) or the Assistant Computer Protection Program Manager (ACPPM). This involvement affords early determination of classifications and levels of protection required for the software product.

If a software product under development processes sensitive personal information, appropriate safeguards must be established to protect the information from accidental disclosure.

Implement applicable security procedures to assure data integrity and protection from unauthorized disclosure, particularly during development efforts. The organization that owns the data defines the data classification. The project team must be aware of all the types of data and of any classified or proprietary algorithms used in the software product.

Procedure: Use the following procedure to determine computer security requirements.

1. Identify the types of data that will be processed by the software product.

2. Determine preliminary data protection requirements.

    a. For software products processing classified information refer to DOE 5639.6, CLASSIFIED COMPUTER SECURITY PROGRAM, September 15, 1992, attachment III, page III-19, paragraph 5.c.(4) - Applications Software.

    b. For software products processing unclassified information, refer to DOE HEADQUARTERS UNCLASSIFIED COMPUTER PROTECTION PLAN, dated December 1993.

    c. For software products processing sensitive information refer to Chapter 5 of the DOE HEADQUARTERS UNCLASSIFIED COMPUTER PROTECTION PLAN.

    d. For software products processing sensitive personal information, contact the Freedom of Information Office for coordination and assistance in complying with DOE 1800.1A, PRIVACY ACT.

Procedure, continued:

e. For software products that are considered to be mission essential refer to paragraph 5.4.1.3 and Chapter 8 of DOE HEADQUARTERS UNCLASSIFIED COMPUTER PROTECTION PLAN.

3. Coordinate with the owner of the host platform to identify existing supporting computer security controls, if applicable.

4. Incorporate security requirements into the Software Requirements Specification.

Work Product: Maintain a record of all security and access requirements. Save for incorporation into the Software Requirements Specification. Place a copy of the security and access requirements in the Project File.

Sample Access Control Questions: The following list provides sample questions that can be used to help define the access controls for the software product.

- What access restrictions are placed on the users by their organization or programmatic office?
- What are the audit and other checking needs for the software product?
- What separation of duties, supervisory functions related to control, operating environment requirements, or other functions will impact the software product?
- What measures will be used to monitor and maintain the integrity of the software product and the data from the user's viewpoint?

Sample Security Requirement: *The application must maintain a record of all user access attempts sorted by authorized and unauthorized users.*

Review Process: Conduct structured walkthroughs as needed to ensure the necessity, testability, accuracy, and completeness of the computer security and access requirements.

References:

- DOE Headquarters Computer Protection Plan (CPP) for Unclassified Systems, February 1995, describes the unclassified computer security program for Headquarters organizations.
- DOE Order 1360.2B, UNCLASSIFIED COMPUTER SECURITY PROGRAM, provides guidance for organizations to implement a computer security program for sensitive information.
- DOE Order 5639.6, CLASSIFIED COMPUTER SECURITY PROGRAM, provides guidance for classified systems.
- DOE 1800.1, PRIVACY ACT

Task: **4.3.8**
Define Backup and Recovery Requirements

Description: Develop the requirements for data backup, recovery, and operation startup for the software product in conjunction with the site authority for continuity of operations. If a software product has been defined as mission essential, a Continuity of Operations Plan must be developed. A checklist is provided in *Exhibit 4.3-1, Checklist for Identifying Mission-Essential Software,* to determine if the software is mission essential.

Work Product: If a software product is determined to be mission essential, a Continuity of Operations Plan must be developed. If the software product is not mission essential, a continuity of operations statement is required. Two samples of continuity of operations statements that are appropriate for software that is not mission essential are provided after the checklist. Place a copy of the Continuity of Operations Statement or Plan in the Project File.

Review Process: Conduct structured walkthroughs as needed to assure the necessity, testability, accuracy, and completeness of the backup and recovery requirements.

Reference: *Disaster Recovery Program Guidelines;* Department of Energy; Office of Information Resource Management; Policy, Plans, and Oversight, July 1991.

Exhibit 4.3-1. Checklist for Identifying Mission-Essential Software

The checklist is intended to be used to help identify software products that are mission essential. If a "yes" answer is selected for one or more of the criteria, the software product is mission essential and a Continuity of Operations Plan must be developed.

	Criterion	Yes	No
1	Inability to perform function adversely affects national security.		
2	Inability to perform function adversely affects safety of individuals.		
3	Needed for military effort and civil defense activities during a national emergency.		
4	Needed for mobilization and protection of material and manpower during national emergency.		
5	Function required for maintenance of public health, safety, and order.		
6	Maintains records essential to preservation of legal rights.		
7	Large financial loss incurred with inability to perform functions.		
8	Large expense incurred if performing function by other means.		
9	Primary repository of information reported to Congress or other agencies.		
10	Critical for compliance with federal regulatory requirements.		
11	Sole source of data unobtainable by other means, or not easily recreated.		

Mainframe Sample Statement:

The backup and recovery of the Human Resources Information System (HRIS) comes under the umbrella of the Virtual Machine system backup and recovery procedure. The Virtual Machine system is backed up daily by VMBACKUP under the control of VMSCHEDULE, an automatic job scheduler software package. It is possible to reconstruct HRIS to its state just prior to any system crash by restoring the data base using VMBLIST, another software package. Additionally, HRIS has a software utility called CALLADM, which, with the appropriate parameters, can provide backup and restore capabilities at the directory record level. This utility is fully documented in Appendix F of the HRIS Installation and Reference Guide. The Virtual Machine system is backed up once a week and the tapes are vaulted offsite.

Microcomputer Sample Statement:

The Human Resources Information System (HRIS) is backed up daily by the HRIS data base administrator, using HRIS system utilities. Both the data base and the HRIS log files are backed up. If there is a media failure, it is possible to reconstruct the data base to its state as of the most recent backup of the log files, using an automated procedure. It is possible to perform backups while HRIS users are connected and transactions are in progress, as well as when they are not. HRIS has a Continuity of Operations Plan that allows its users to continue operations using a server in Germantown, Maryland should the local area network (LAN) become unavailable for a significant period. HRIS will be included in the Continuity of Operations Plan being developed for the Human Resources and Administration LAN. This Continuity of Operations Plan is scheduled for completion during the current fiscal year.

Task: **4.3.9**
Define Data Requirements

Description: Data requirements identify the data elements and logical data groupings that will be stored and processed by the software product. The identification and grouping of data begins during the Requirements Definition Stage and is expanded in subsequent stages as more information about the data is known.

Work Product: The major output of the data requirements identification process is a data dictionary. A data dictionary provides an ordered set of definitions about data inputs and outputs, and data stores. In the Requirements Definition Stage, the data dictionary contains a minimum amount of information about data elements such as definitions of the entities, how the data are stored, and data flows to or from other applications. The data dictionary is refined during the design stages as data elements are documented in more detail, and the logical groupings of data elements are formed into interrelated tables or record descriptions.

Work Product: Maintain a record of all data requirements. Save for incorporation into the Software Requirements Specification. Place a copy of the data requirements in the Project File.

Sample Data Requirement: *Records imported from HRIS and ABCD must be matched to the application using the employee's social security number.*

Review Process: Conduct structured walkthroughs as needed to ensure the necessity, testability, accuracy, and completeness of the data requirements.

Task: **4.3.10**
Define Implementation Requirements

Description: Describe the requirements anticipated for implementing the software product (e.g., user production cycle). The high-level implementation requirements are identified early in the lifecycle to support decisions that need to be made for the software engineering approach. The implementation requirements are expanded into a full implementation approach during the design stages.

The following paragraphs provide highlights of some of the implementation requirements that need to be considered.

Operating Environment: Identify any capacity restrictions on the existing hardware or software that needs to be addressed and identify any hardware or software that needs to be acquired (e.g., communication hardware, file servers, off-the-shelf software, network interface cards, and LAN utilities). If hardware or software must be acquired, identify the necessary acquisition activities. These activities include preparing specifications, estimating costs, scheduling procurement activities, selection, installation, and testing.

Conversion: Identify requirements for converting data from an existing or external application to the new software product. Consider requirements for data entry, data protection, computer time, conversion programs, personnel, and other resources that will be needed. Also identify the requirements for the conversion of software, if necessary. Implementing a new application may involve converting software from one environment to another, or modifying software to interface with other applications. Include requirements for testing the conversion process and validating that it was successfully accomplished.

Installation: Identify the installation requirements for any new hardware, operating system, or software. For hardware installations, consider environmental factors such as air conditioning, power supply, and security requirements. For software installations, consider proprietary software such as data base management systems. For application software, consider the installation of the application's programs, parallel operation of the old and new applications, or the cutover from a test to a production environment. Hardware and software installation must be coordinated with the work cycles of the user organization to create a minimum of disruption, and to assure that data are available as needed. Installation must be scheduled to assure that, when data conversion is necessary, the needed data are protected.

Training: Identify the specific training needs for various categories of users and administrators. Also identify training requirements for personnel time, computer time, training facilities, and training data base(s).

Documentation: Identify requirements for the development and distribution of operational documentation for software support personnel and user documentation. Operational documentation may include job control procedures and listings, operational instructions, system administration responsibilities, archiving procedures, and error recovery. User documentation includes the users manual, step-by-step instructions, online documentation, and online help facilities.

Work Product: Maintain a record of all implementation requirements. Save for incorporation into the Software Requirements Specification. Place a copy of the implementation requirements in the Project File. This information will also be used to develop an Implementation Plan in the Functional Design Stage.

Sample Conversion Requirement: *All Julian dates found in the extract files must be converted to Gregorian dates.*

Review Process: Conduct structured walkthroughs as needed to ensure the necessity, testability, accuracy, and completeness of the implementation requirements.